УДК 004:652.6:512.58

Н. Н. СЛИСАРЕНКО

ОБ АЛГЕБРАИЧЕСКОМ ПОДХОДЕ К ОПИСАНИЮ ОБОБЩЕННЫХ СУЩНОСТЕЙ

В статті пропонується нова методика розробки логічних моделей складних інформаційних систем з використанням узагальнених сутностей. Показано, що узагальнені сутності можуть бути представлені категоріями, а у більш загальному випадку рефлексивними категоріями.
Застосування узагальнених сутностей у логічних моделях і реалізація їх на рівні деякої СУБД дозволить покращити "масштабність" програмного забезпечення.

Темпы развития современного общества определяются уровнем информатизации. Одним из основных факторов, оказывающих влияние на процесс информатизации, является наличие современной технологии и инструментальных средств проектирования, сложных информационных управляющих систем, под которыми понимаются системы сбора, хранения, обработки информации, независимо от функционального назначения.

Проектирование сложных информационных управляющих систем является процессом, требующим больших затрат ресурсов и времени, а также привлечения большого количества высококвалифицированных специалистов.

Масштабы работ по информатизации современного общества привели к тому, что ещё в 70-х годах в развитых странах были разработаны и приняты на правительственном уровне Государственные программы создания и развития индустриальных технологий проектирования, такие как SSADM , IDEF, Merice, Dafhe, NIAM. Как правило, данные средства применяются для проведения анализа и реорганизации бизнес процессов, предоставляя средства описание существующих бизнес процессов на предприятии.
После создания эффективной схемы функционирования предприятия возникает следующий вопрос: как представить данное описание в виде логической модели данных (в данном случае ER модели), которую можно будет реализовать в конкретной СУБД. К сожалению, данный процесс плохо формализуется и поэтому не полностью автоматизирован, но результатом подобного процесса является, как правило, достаточно большая и непрозрачная для не специалиста ER модель.

На практике, при реализации задач автоматизации часто возникает проблема изменения требований к системе. Как правило, при любом радикальном вмешательстве в уже существующую систему, её приходится перерабатывать практически с нуля. Это натолкнуло на мысль о необходимости использования такой схемы разработки, которая при включении и изменении элементов в общую структуру ER модели, сохраняла бы её функциональность.

На данный момент для построения ER моделей, наиболее часто используется разработанная для армии США и сейчас широко используемая в государственных учреждениях и промышленных корпорациях нотация IDEF1X.

Логическая модель этой идеологии включает в себя два основных уровня логической модели, отличающихся друг от друга глубиной представления информации о данных:

1. Диаграмма “сущность - связь” - модель верхнего уровня, включающая в себя только сущности и связи предметной области.
2. Полная атрибутивная модель.

При разработке реальных систем с использованием нотации IDEF1X возникает прямая зависимость прозрачности модели от её размеров. Чем больше схема, тем она становится менее прозрачной. Проблема заключается в том, что, опираясь на главное правило построения сущностей, любое явление или объект может быть представлено в виде только одной сущности. Такое определение сущности целесообразно на стадии законченной, уже однозначной модели. На практике же, в процессе построения модели, часто приходится частично или полностью изменять структуру модели, последовательность и характер связей между сущностями и сами сущности в зависимости от новых требований. Было бы более целесообразно располагать таким механизмом, который позволил бы работать с сущностями более высокого уровня абстракции, которые в процессе разработки модели могли бы быть детализированы более подробно с учётом новых требований к системе.

Для решения данной задачи предлагается расширение ER моделей, путем использования обобщенных сущностей (ОС).

Методика формирования ОС основывается на двух принципах: на деятельностном принципе и принципе включения сущностей в ОС (абстрактное – конкретное либо часть - целое).

Применение подобной схемы построения обобщенной ER модели позволит значительно повысить уровень абстракции модели, что крайне необходимо на начальных этапах разработки, в частности на этапе общения с заказчиком. Также эта схема позволит значительно увеличить прозрачность ER модели, за счет укрупнения элементов модели в ОС. Кроме того, расширенная ER модель позволит работать с элементами ОС таким образом, что это не будет пагубно отражаться на модели в целом, другими словами, повысится общая масштабируемость ER модели.

В данной работе показывается, что ОС можно представить в виде алгебраической структуры - категории. Категорный подход к представлению ОС является естественным развитием реляционного подхода к БД, начало которого было заложено в работах Е. Кодда [1].

Сначала рассмотрим основные понятия и определения категорий.

Понятия категории и функтора были введены в математику С. Эйленбергом и С. Маклейном в 1944 году [2]. Эти понятия широко используются во многих областях математики. В последние десятилетия с развитием теоретических исследований в компьютерных науках они используются при описании абстрактных структур данных в программировании, в реляционных СУБД при описании схем баз данных, в работах по созданию БД понятий [3].

Определение 1. Говорят, что задана категория ζ, если определены:

1. Класс Obζ элементов, называемых объектами категории: A, B, C, …;
2. Множество Morζ морфизмов (отображений, стрелок): f, g, h, …;

Чтобы показать, что f есть отображение будем писать $A \xrightarrow{f} B$ (или $f\colon A \to B$);

3. Для каждого объекта $A \in$ Obζ имеется тождественное отображение которое обозначается 1A, такое что $1A\colon A \to A$.
4. Для каждой пары отображений (f,g) $A \xrightarrow{f} B \xrightarrow{g} C$ существует композиция отображений $A \xrightarrow{g\_следует\_f} C$, которую обозначим через $g \circ f$, и которая удовлетворяет следующим законам:

(a) закон идентичности:

$$\text{если } f\colon A \to B \text{, то } 1B \circ f = f \text{ и } f \circ 1A = f .$$

(b) закон ассоциативности:

$$\text{если } A \xrightarrow{f} B \xrightarrow{g} C \xrightarrow{h} D \text{ ,то } (h \circ g) \circ f = h \circ (g \circ f) .$$

В определении категории (1) ничего не говорится о теоретико-множественной структуре объектов, поэтому мы не можем в общем случае работать с «элементами» таких объектов.

Все основные общекатегорные конструкции и их приложения к конкретным категориям формулируются преимущественно в терминах морфизмов и их композиции. Удобным языком для таких формулировок является язык диаграмм. Например, вместо того чтобы говорить, что у нас имеются четыре объекта A, B, C, D и четыре морфизма $f\colon A \to B$, $g\colon B \to D$, $h\colon A \to C$ и $d\colon C \to D$, причем $g \circ f = d \circ h$, говорят, что задан коммутативный квадрат (см. рис 1):

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ C \arrow[r, "d"] & D \end{tikzcd}$$

Рис. 1.

Коммутативность здесь - это равенство g f = d h, которое означает, что два пути вдоль стрелок от A к D приводят к одному и тому же результату.

Более того, диаграмма - это ориентированный граф, вершины которого являются объектами ζ, а ребра морфизмами.

Частным случаем диаграмм являются комплексы.

Комплекс – конечная или бесконечная последовательность объектов и стрелок $A \to B \to C \to D\ldots$

Часто объекты и морфизмы, входящие в комплексы, индексируются некоторым отрезком целых чисел

$$a0 \xrightarrow{f_0} a1 \xrightarrow{f_1} a2 \xrightarrow{f_2} a3 \xrightarrow{f_3}$$

Рассмотрим важное понятие функтора.

Определение 2. Функтор – это отображение из одной категории в другую, сохраняющее категорную структуру.

Функтором F из категории ζ_1 в категорию ζ_2 называется функция, ставящая в соответствие:

1. Каждому объекту $a \in \zeta_1$ - некоторый объект $F(a) \in \zeta_2$;

2. Каждой стрелке $f\colon a \to b \in \zeta_1$ стрелку $F(f)\colon F(a) \to F(b) \in \zeta_2$ такую, что:

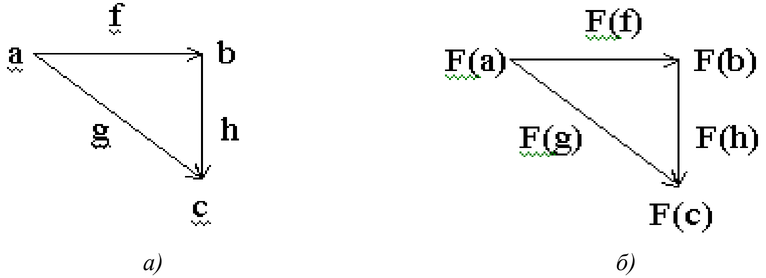
(a) $F(1a) = 1F(a)$ для каждого объекта $a \in \zeta_1$ т.е. единичной стрелке, соответствующей объекту a, сопоставляется единичная стрелка, соответствующая объекту F(a);

(b) $F(g \circ f) = F(g) \circ F(f)$ для любых g и f , для которых определена композиция $g \circ f$.

Последнее условие означает, что F образ композиции двух стрелок является композицией их F образов, т.е. всякий раз, когда диаграмма коммутативна в ζ_1 (рис. 2, *а*), соответствующая диаграмма коммутативна в ζ_2 (рис. 2, *б*). Для того чтобы выразить тот факт, что F есть функтор из ζ_1 в ζ_2, будем писать

$$F\colon \zeta_1 \to \zeta_2 \text{ или } \zeta_1 \xrightarrow{F} \zeta_2.$$

Таким образом, функтор – это отображение, сохраняющее единичные стрелки и композиции.



а) *б)*

Рис. 2.

Рассмотрим ОС. Обобщенная сущность $\overline{E}$ представляется в логической модели «сущность-связь» (ER-модели), последовательностью $E_1,\ldots, E_n$ с заданными связями между ними $R_{ij}, \ldots, R_{n-1,n}$.

Уточним теперь какими свойствами обязательно обладает совокупность сущностей E_i и связей R_{ij} (для i=1, n) независимо от того, какая ОС ими представляется. Так как связям R_{ij} в конкретных реализациях соответствуют отображения одних множеств в другие, то выполняется следующее свойство:

если R_{12}: $E_1 \rightarrow E_2$ и R_{23}: $E_2 \rightarrow E_3$ такие отображения, что $(R_{12})\_ = \_(R_{23})$, где

( )_ - операция выделения области значений;

_( ) - операции выделения области определений, то строится новое отображение $R_{23} \circ R_{12}$: $E_1 \rightarrow E_3$, которое назовем композицией отображений R_{12} и R_{23}.

Композиция отображений обладает свойством ассоциативности. Это значит, что для любых отображений R_{12} и R_{23} и любого отображения R_{34}: $E_3 \rightarrow E_4$ выполняется равенство:

$$R_{34} \circ (R_{23} \circ R_{12}) = (R_{34} \circ R_{23}) \circ R_{12}.$$

Кроме этого, для любой сущности E ОС можно ввести тождественное отображение id(E): E→E композиция которого с любыми отображениями f: $E_i \rightarrow E$ и g: $E \rightarrow E_j$ дает следующие равенства:

$$id\,(E) \circ f = f \text{ и } g \circ id\,(E) = g$$

Как отмечалось ранее (определение 1, п. 3 и п. 4), математическая структура, обладающая операциями типа операции “композиция” со свойствами (определение 1, п. 1 и п. 2), является категорией. Объектами этой категории являются сущности ОС, а морфизмами – отношения между сущностями ОС. Обозначим эту категорию через С.

Для указания элементов объектов категории в теории категорий используется следующий прием.

Вводится особый объект точка, который сокращенно обозначается через I, обладающий следующим свойством:

для каждой сущности Е из категории С существует единственный морфизм из Е в I, который обозначается через I (E): E→I. Это свойство полностью характеризует объект точка. В теории категории его называет финальным объектом. Полезно также иметь пустой объект (сущность ø), который задается следующим свойством: для любой сущности категории С множество всех морфизмов С(ø, Е) состоит из единственного морфизма ø (E): ø →E. В теории категорий такой объект называют инициальным.

В дальнейшем будем предполагать, что рассматриваемая здесь категория С имеет сущности ø и I.

Категория С называется непротиворечивой, если область ø неизоморфна области I. Соответственно, если ОС $\overline{E}$ представлена непротиворечивой категорией, то представление называется непротиворечивым.

Пусть E некоторая сущность из категории C. Известно, что элементы объекта задаются морфизмами e: $I \to E$ из области точка в область E. Множество морфизмов C (I,E) соответствует множеству всех известных элементов сущности E в категории C.

Если f: $E_1 \to E_2$ морфизм, то он индуцирует отображение известных элементов сущности E_1 в известные элементы сущности E_2. Действительно, пусть e: $I \to E_1$ произвольный элемент сущности E_1, тогда композиция морфизмов $f \circ e$: $I \to E_2$ элемент сущности E_2.

В основе категорного подхода к представлению ОС так же как и в основе реляционного подхода в моделях баз данных лежит некоторый набор операций, называемых категорными [4]. Категорные операции позволяют по одним сущностям и преобразованиям (морфизмам) строить (выражать) другие сущности и преобразования.

Пусть $\mathcal{F}$ – множество морфизмов, а $\mathcal{E}$ – множество сущностей категории C.

Приведем операции, которые уже встречались в ходе рассмотрения ОС.

1. Область_определения: $\mathcal{F} \to \mathcal{E}$. Обозначается _( ). Для заданного морфизма f: $E_1 \to E_2$ результатом будет сущность E_1.

2. Область_значений: $\mathcal{F} \to \mathcal{E}$. Обозначается ( )_. Для заданного морфизма f: $E_1 \to E_2$ результатом будет сущность E_2.

3. Тождественый_морфизм: $\mathcal{E} \to \mathcal{F}$. Сущности E сопоставляет морфизм 1E.

4. Композиция: $\mathcal{F} \times \mathcal{F} \to \mathcal{F}$. Обозначается символом $\circ$. Операция композиция сопоставляет паре морфизмов (f_1, f_2) третий морфизм обозначаемый $f_2 \circ f_1$, если _(f_2)=(f_1)_.

5. Операция _IE: $\mathcal{E} \to \mathcal{F}$. Эта операция сущности E сопоставляет морфизм IE.

6. Операция_øE: $\mathcal{E} \to \mathcal{F}$. Эта операция сущности E сопоставляет морфизм øE.

Как видно из приведенных операций, только четвертая создает что-то новое, а остальные извлекают из категории известные элементы ОС.

7. Операция_ТМП: $\mathcal{E} \times \mathcal{E} \to \mathcal{E} \times \mathcal{F}$.

Очень важным понятием в теории категорий является теоретико-множественное произведение (ТМП) двух объектов.

Пусть E_1 и E_2 объекты категории C. Декартово произведение $E_1 \times E_2$ – это новый объект вместе с морфизмами (проекциями) pr_1: $E_1 \times E_2 \to E_1$ и pr_2: $E_1 \times E_2 \to E_2$, обладающие следующими свойствами: для любой пары морфизмов f_1: $E \to E_1$ и f_2: $E \to E_2$ есть морфизм (f_1, f_2): $E \to E_1 \times E_2$, композиция которого с проекциями pr_1 и pr_2 дает равенства:

$$pr_1 \circ (f_1, f_2) = f_1 \text{ и } pr_2 \circ (f_1, f_2) = f_2,$$

и, наоборот, любой морфизм вида $f: E \to E_1 \times E_2$, представляется в виде

$$f = (pr_1 \circ f, pr_2 \circ f).$$

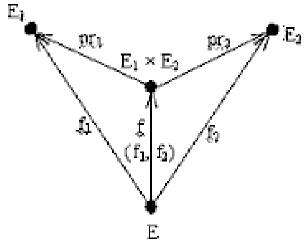


Рис.3

В этом определении декартовое произведение объектов задается следующим набором операций: произведением объектов $E_1 \times E_2$, проекциями pr_1 (E_1, E_2), pr_2 (E_1, E_2) и произведением морфизмов (f_1, f_2), которые по областям E_1, E_2 и морфизмам f_1, f_2 строят соответствующие объект и морфизмы.

Приведенных операций вполне достаточно для ведения ОС и извлечения из неё информации. Однако в практических приложениях для различных ОС могут понадобиться и другие полезные операции. Целесообразно было бы унифицировать систему операций так, чтобы набор операций был конечным и единым для различных ОС.

Удовлетворить этим казалось бы противоречивым требованиям к системе операций, предъявляемым задачами представления ОС, оказалось возможно за счет введения конечного набора особых операций над объектами и морфизмами категории С, который позволяет иметь описание самой категории внутри этой же категории С. Такие категории называются рефлексивными [5].

Пусть в категории С дополнительно заданы две области Name_o – множество имен объектов и Name_m – множество имен морфизмов (преобразований) вместе с финальным объектом I категории С, а также с преобразованиями между ними с соотношениями, которые входят в определение понятия категории, так что относительно этих преобразований множества элементов областей Name_o и Name_m образуют категорию, которая называется категорией имен.

Введем необходимые морфизмы этой категории:

e: Name_o→Name_m отображает имя объекта (E) категории С в имя его тождественного морфизма 1E:E→E;

Name_dom: Name_m→Name_o отображает имя морфизма в имя объекта, являющегося его областью определения, т.е. для морфизма $f: E_1 \to E_2$ отображает имя f в имя E_1.

Name_codom: Name_m→Name_o отображает имя морфизма в имя объекта, являющегося его областью значений, т.е. для морфизма f: $E_1 \to E_2$ отображает имя f в имя E_2.

По смыслу этих морфизмов между ними должны выполняться соотношения:

$$\text{Name\_dom} \circ e = id\ (\text{Name\_o}),\ \text{Name\_codom} \circ e = id\ (\text{Name\_o}),$$

которые означают, что имена области определения и области значений для имени тождественного морфизма совпадают с именем объекта этого морфизма.

Дополним категорию имен морфизмом композиции имен. Обозначим через E объект в категории C, который получается операцией уравнивания пары морфизмов Name_dom ○ pr_1 и Name_codom ○ pr_2 из произведения объектов Name_m× Name_m в Name_o, где pr_1 – проекция произведения Name_m× Name_m на первый сомножитель, а pr_2 – проекция на второй сомножитель. То есть E это подобъект в Name_m× Name_m, состоящий из тех пар имен морфизмов, у которых имя области определения первой компоненты пары совпадают с именем области значений второй.

Пусть в категории C задан морфизм вида comp_name: Name_o→Name_m, который называется композицией имен морфизмов. Если в Name_m были f1: $E_1 \to E_2$, f2: $E_2 \to E_3$, и не было имени f_3: $E_1 \to E_3$, то по морфизму comp_name, задавая E_2, получим морфизм f_3: $E_1 \to$E3 в Name_m.

Предполагается также, что для морфизмов comp_name, Name_dom, Name_codom, e выполняются соотношения являющиеся аналогами аксиом категорий. Следовательно, объект (Name_o, Name_m, comp_name, Name_dom, Name_codom, e) является внутренней категорией категории C: Тогда финальный объект I категории C , множество элементов объектов Name_o и Name_m и операции на них, индуцированные морфизмами comp_name, Name_dom, Name_codom, e, по определению внутренней категории образуют малую категорию.

Эту категорию имен назовем C_{name} .

Определение 3. Категория C вместе с внутренней категорией Cname , называется категорией с категорией имен [Д] .

Определение 4. Категория C вместе с внутренней категорией Cname называется рефлексивной категорией C^R, если существуют функторы

F_n: $C \to C_{name}$ и F_d: $C_{name} \to C$, такие что их композиция $F_d \circ F_n = 1C$, является тождественным функтором на C.

Функтор F_n называется функтором именования, функтор F_d – денотатом (т.е. функтором разименования).

Функтор F_n по каждой области E категории C строит элемент:

F_n (E): I→Nam_o в области Name_o категории C_{name} , а по каждому морфизму категории C строит элемент F_n (f): I→ Name_m в области Name_m категории C_{name}.

При этом на функторы F_n и F_d накладываются соотношения:

$$F_d(F_n(E)) = E \text{ и } F_d(F_n(f)) = f$$

для любых объектов E и морфизмов f в категории C, т.е. Denote - это обратная функция по отношению к F_n, что было отмечено выше:

$$F_d \circ F_n = 1C.$$

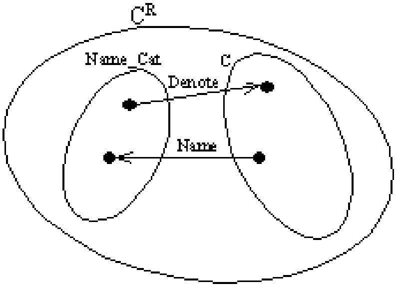


Рис. 4

Таким образом, рефлексивная категория C^R , являющаяся алгебраической моделью ОС, состоит из категорий C и C_{name} и двух функторов F_n и F_d.

$C^R = (\mathcal{E}, \mathcal{R}, \mathcal{OER}, \mathcal{N}c, \mathcal{N}м, \mathcal{O}см, F_n, F_d.)$, где

$\mathcal{E}$ – множество сущностей ОС;

$\mathcal{R}$ – множество морфизмов (связей) ОС;

$\mathcal{OER}$ – операции в категории С;

$\mathcal{N}c$ – множество имён сущностей;

$\mathcal{N}м$ – множество имён морфизмов;

$\mathcal{O}см$ –операции в категории C_{name};

F_n – функтор именования : $C \to C_{name}$;

F_d – функтор разыменования: $C_{name} \to C$.

Использование категории C^R для представления ОС позволяет отражать знания о структуре всей категории, обращаясь к описанию внутри категории.

Дополнительно к этому, использование категорных операций категории C^R позволяет строить новые операции над объектами и преобразованиями категории С, вводя новые преобразования лишь над именами, а с помощью функторов F_n и F_d находить соответствующие им объекты и морфизмы.

Применение ОС в логических моделях прикладной области и реализации категории CR на уровне некоторой СУБД позволит повысить «масштабируемость» ПО до уровня включения (исключения) сущностей из последовательности сущностей ОС.

Список литературы*:* ***1.*** *Codd E.F.* A relational model for large shared data banks. Comm. of ACM 13, 6, 1970, p.377-387. ***3.*** *Бениаминов Е. М.* Система представления и обработки понятий основанная на алгебраическом (категорном) подходе. Труды II Всесоюзной конференции "Искусственный интеллект - 90", т.2, 1990, с.8-11. **4.** *Голблатт Р.* Топосы. Категорный анализ логики. М.: Мир, 1983. *5.Джонстон П.Т.* Теория топосов. М.: Наука, 1986. 3.

Поступила в редколлегию 16.11.05